MULTIVARIATE PROBABILISTIC TIME SERIES FORECASTING VIA CONDITIONED NORMALIZING FLOWS

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann & Roland Vollgraf
Zalando Research

PRESENTED BY NITIN KUMAR

Outline

Introduction

- Probabilistic forecasting means estimating the probability distribution of a time series' future given its past
- A Multivariate time series has more than one time-dependent variable.
- Each variable depends not only on its past values but also has some dependency on other variables.
- This dependency is used for forecasting future values.

Time	Temperature	cloud cover	dew point	humidity	wind
5:00 am	59 °F	97%	51 °F	74%	8 mph SSE
6:00 am	59 °F	89%	51 °F	75%	8 mph SSE
7:00 am	58 °F	79%	51 °F	76%	7 mph SSE
8:00 am	58 °F	74%	51 °F	77%	7 mph S
9:00 am	60 °F	74%	51 °F	74%	7 mph S

Introduction

- **Why Probabilistic Modeling**?
- Without it, the importance of the forecast in regions of low noise (small variance around a mean value) versus a scenario with high noise cannot be distinguished.
- Point estimation models ignore risk stemming from this noise. It is of particular interest in many cases such as making (business) decisions.

- **Why Multivariate Modeling**?
- Individual time series, in many cases, are statistically dependent on each other.
- Models need the capacity to adapt to this in order to improve forecast accuracy.
- E.g. A traffic flow in a network of streets as measured by occupancy sensors. A disruption on one street will also ripple to occupancy sensors of nearby streets.

Introduction

- The Paper proposes end-to-end trainable autoregressive deep learning architectures for probabilistic forecasting that explicitly models multivariate time series and their temporal dynamics by employing a normalizing flow.
- Models can scale to thousands of interacting time series.
- Methods adapt to a broad class of underlying data distribution on account of using a normalizing flow.
- It also uses transformers which makes it highly efficient due to the parallel nature of attention layers while training.

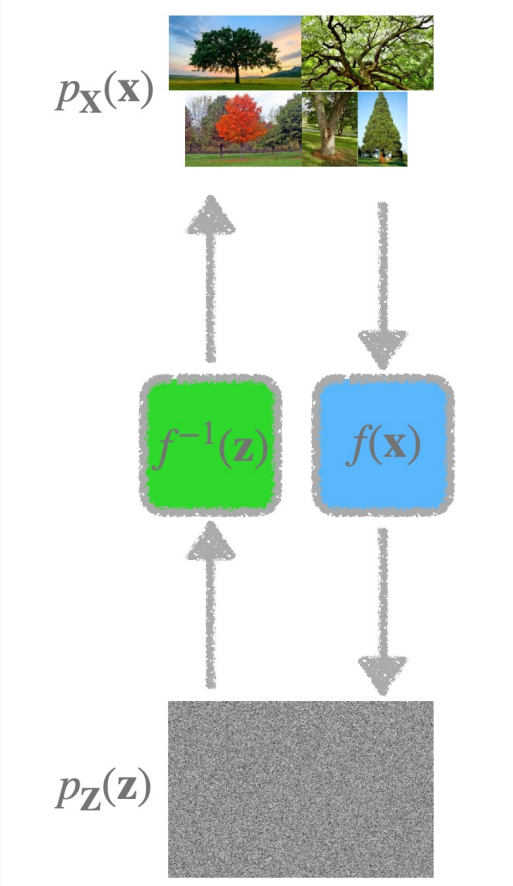
Use cases

Forecast :

- Energy consumption of individual households
- Load for servers in a data center
- Demand for all products that a large retailer offers
- Weather based on the past weather data
- Stock market prediction based on the past trend

Background

Density Estimation via Normalizing Flows:



Background

Density Estimation via Normalizing Flows:

$$p_X(x) = p_Z(z) \left| \det \left(\frac{\partial g(z)}{\partial z^T} \right) \right|^{-1}.$$

Given an observed data variable $x \in X$, a simple prior probability distribution p_Z on a latent variable $z \in Z$, and a bijection $f : X \to Z$ (with $g = f^{-1}$), the change of variable formula defines a model distribution on X by

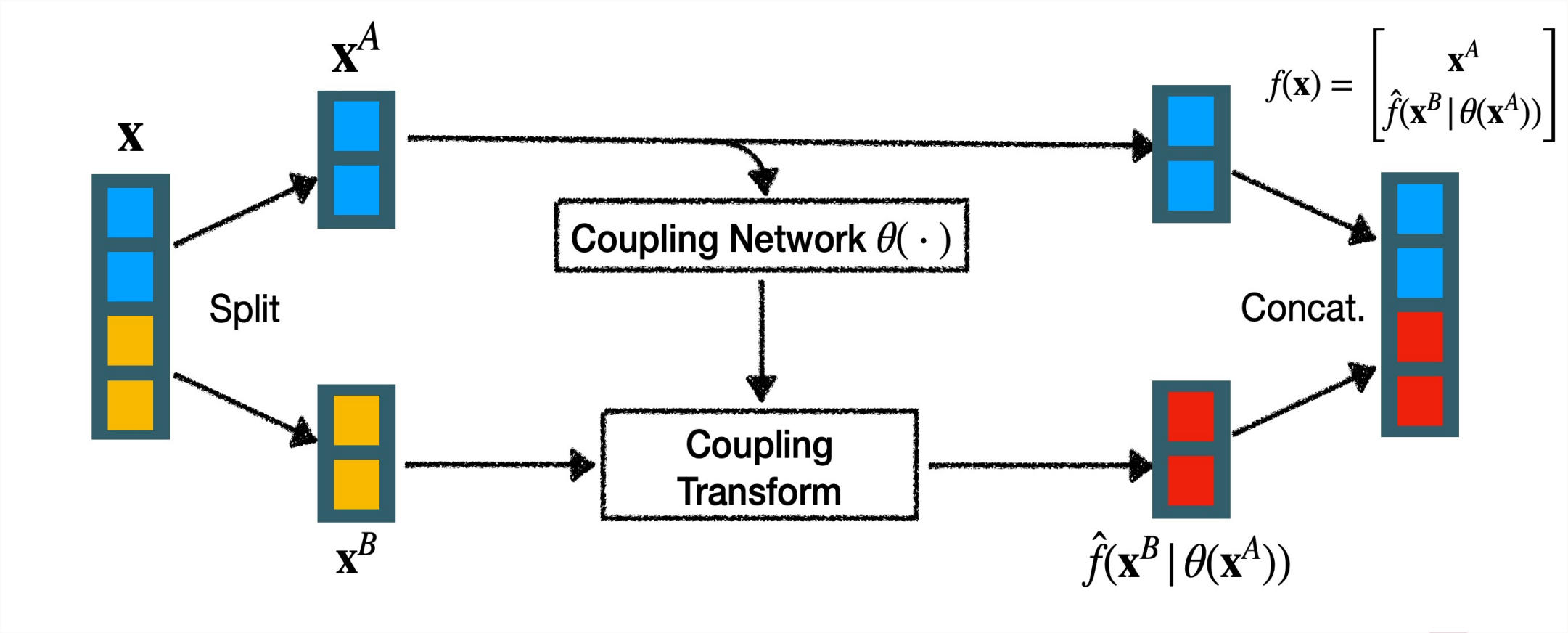
$$p_X(x) = p_Z\big(f(x)\big) \left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right| \tag{2}$$

$$\log\big(p_X(x)\big) = \log\Big(p_Z\big(f(x)\big)\Big) + \log\left(\left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right| \right), \tag{3}$$

where $\frac{\partial f(x)}{\partial x^T}$ is the Jacobian of f at x.

Background

Density Estimation via Normalizing Flows:



Background

Density Estimation via Normalizing Flows:

- **Coupling Layer:**
- Given a D dimensional input x and d < D, the output y of an affine coupling layer follows the equations:

$$y_{1:d} = x_{1:d}$$
$$y_{d+1:D} = x_{d+1:D} \odot \exp\left(s(x_{1:d})\right) + t(x_{1:d}),$$

Background

Density Estimation via Normalizing Flows:

- **Combining Coupling Layers**

The Jacobian determinant of the resulting function remains tractable, relying on the fact that

$$\frac{\partial(f_b \circ f_a)}{\partial x_a^T}(x_a) = \frac{\partial f_a}{\partial x_a^T}(x_a) \cdot \frac{\partial f_b}{\partial x_b^T}\big(x_b = f_a(x_a)\big)$$

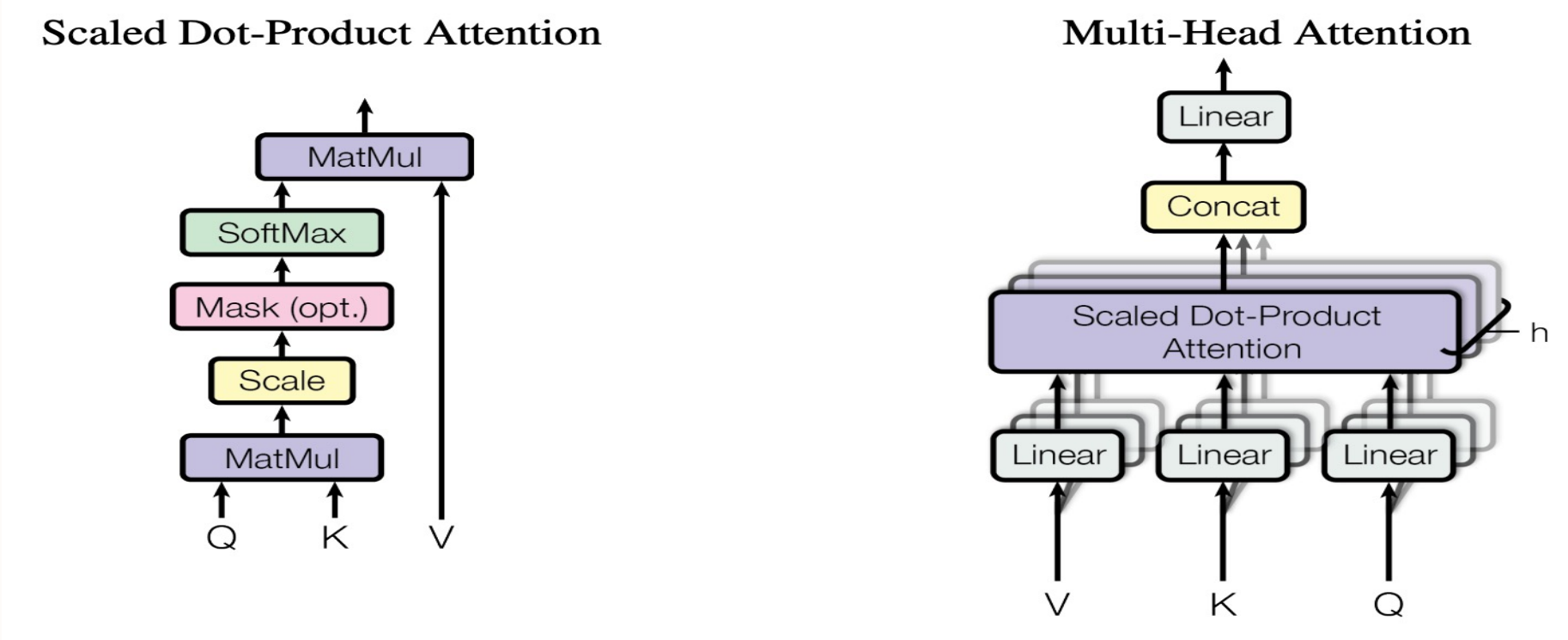
$$\det(A \cdot B) = \det(A)\det(B).$$

Similarly, its inverse can be computed easily as

$$(f_b \circ f_a)^{-1} = f_a^{-1} \circ f_b^{-1}.$$

Background

Transformers:



Temporal Conditioned Normalizing Flows

- Entities of a multivariate time series is denoted by $x_t^i \in \mathbb{R} \text{ for } i \in \{1, \ldots, D\}$
- Features :
  - Scalable model of D interacting time-series x(t).
  - A flexible distribution model on the emissions that allows for capturing and representing higher order moments.
- Modeling is done using Conditional joint distribution at time t of all time series $p_{\mathcal{X}}(\mathbf{x}_t|\mathbf{h}_t;\theta)$ with a flow, e.g. a Real NVP, conditioned on either the hidden state of a RNN at time t or an embedding of the time series up to t − 1 from an attention module.

Temporal Conditioned Normalizing Flows

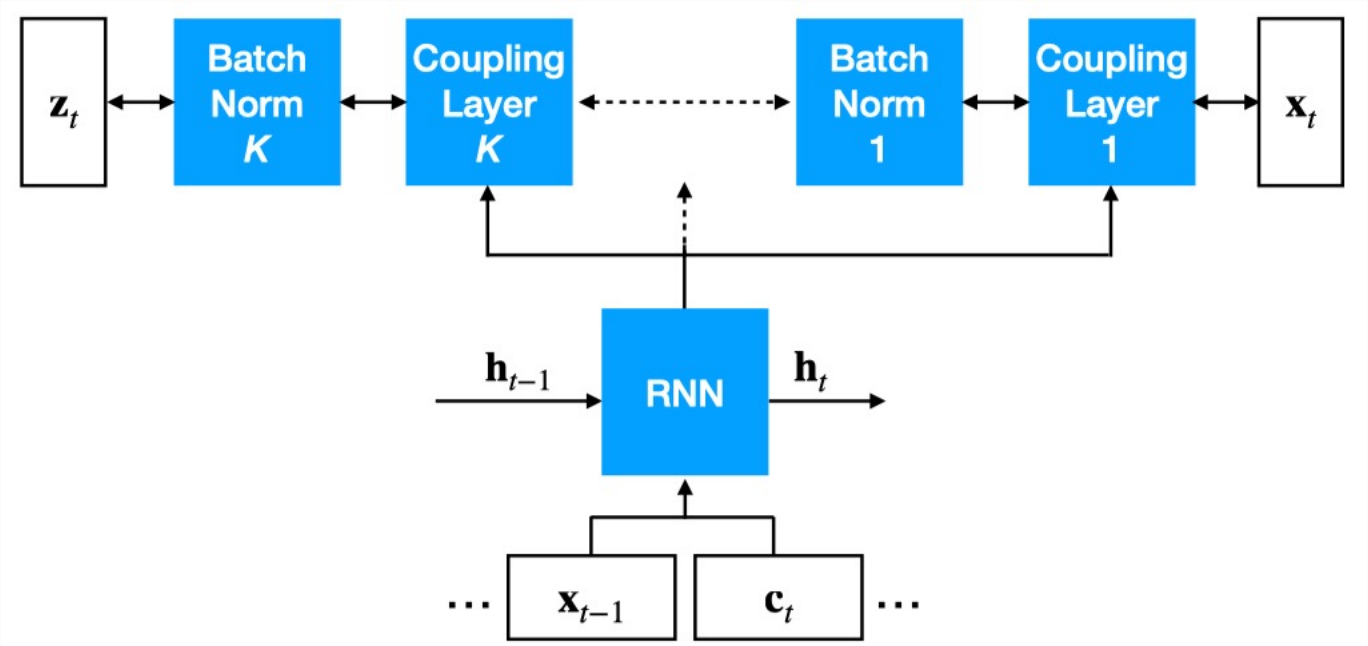


Figure 1: RNN Conditioned Real NVP model

Temporal Conditioned Normalizing Flows

- Case 1: an autoregressive RNN:
  - Hidden state h(t) is updated given the previous time step observation x(t−1) and the current time step's covariates c(t).

$$\mathbf{h}_t = \text{RNN}(\text{concat}(\mathbf{x}_{t-1}, \mathbf{c}_t), \mathbf{h}_{t-1}).$$

$$p_{\mathcal{X}}(\mathbf{x}_{t_0:T} | \mathbf{x}_{1:t_0-1}, \mathbf{c}_{1:T}; \theta) = \Pi_{t=t_0}^{T} p_{\mathcal{X}}(\mathbf{x}_t | \mathbf{h}_t; \theta)$$

where θ denotes the set of all parameters of both the flow and the RNN.

- Case 2: Encoder Decoder using Transformer:
  - Encoder embeds $x_{1:\,t0\,-1}$ and the decoder outputs the conditioning for the flow over $x_{t0\,:T}$ via a masked attention module

Temporal Conditioned Normalizing Flows

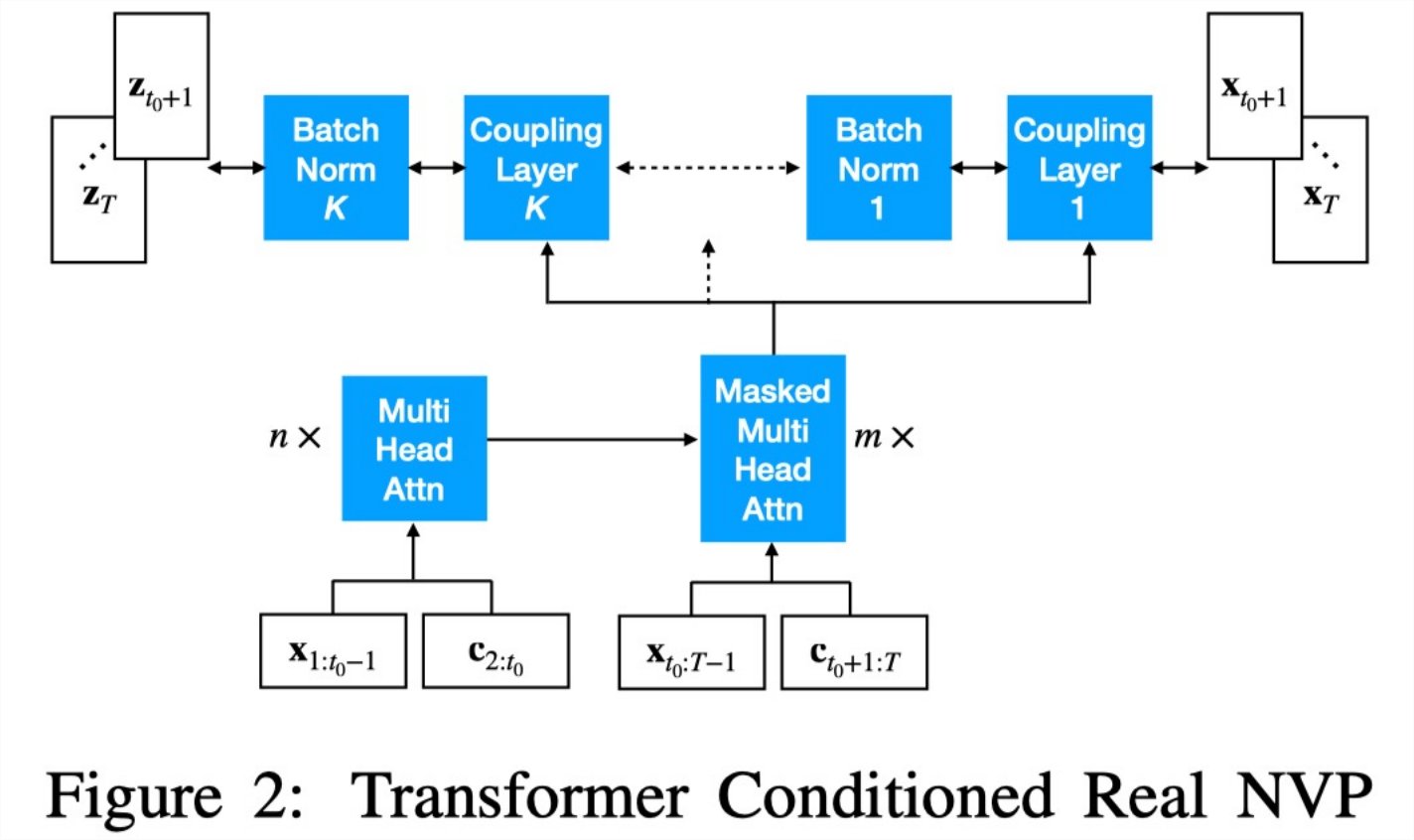


Figure 2: Transformer Conditioned Real NVP

Training

- Given D, a batch of time series, where for each time series and each time step we have $x_t \in R^D$ and their associated covariates c_t , we maximize the log-likelihood via SGD given by:

$$\mathcal{L} = \frac{1}{|\mathcal{D}|T} \sum_{\mathbf{x}_{1:T} \in \mathcal{D}} \sum_{t=1}^{T} \log p_{\mathcal{X}}(\mathbf{x}_t | \mathbf{h}_t; \theta)$$

- Time series $x_{1:T}$ in a batch D are selected from a random time window of size T within training data, and the relative time steps are kept constant.
- This allows the model to learn to conditioning for the prediction length portion of cold-start given only the covariates.

Experiment/Results

Table 3: Test set CRPS_{sum} (lower is better) of classical methods and our `Transformer-MAF` model, where the mean and standard errors of our model are obtained over a mean of 20 runs.

Data set	VAR	VAR-Lasso	GP	GARCH	VES	KVAE	Transformer MAF
Exchange	0.010±0.00	0.012±0.000	0.011±0.001	0.020±0.000	**0.005**±0.00	0.014±0.002	**0.005**±0.003
Solar	0.524±0.001	0.51±0.006	0.828±0.01	0.869±0.00	0.9±0.003	0.34±0.025	**0.301**±0.014
Electricity	0.031±0.00	0.025±0.00	0.947±0.016	0.278±0.00	0.88±0.003	0.051±0.019	**0.0207**±0.000
Traffic	0.144±0.00	0.15±0.002	2.198±0.774	0.368±0.00	0.35±0.002	0.1±0.005	**0.056**±0.001
Taxi	0.292±0.00	-	0.425±0.199	-	-	-	**0.179**±0.002
Wikipedia	3.4±0.003	3.1±0.004	0.933±0.003	-	-	0.095±0.012	**0.063**±0.003

Experiment/Results

Table 4: Test set CRPS comparison (lower is better) of models from Salinas et al. (2019a) and our models LSTM-Real-NVP, LSTM-MAF and Transformer-MAF. The mean and standard errors are obtained by re-running each method three times.

Data set	Vec-LSTM ind-scaling	Vec-LSTM lowrank-Copula	GP scaling	GP Copula	LSTM Real-NVP	LSTM MAF	Transformer MAF
Exchange	0.013±0.000	0.009±0.000	0.017±0.000	**0.008±0.000**	0.010±0.001	0.012±0.003	0.012±0.003
Solar	0.434±0.012	0.384±0.010	0.415±0.009	0.371±0.022	**0.365±0.02**	0.378±0.032	0.368±0.001
Electricity	0.059±0.001	0.084±0.006	0.053±0.000	0.056±0.002	0.059±0.001	**0.051±0.000**	0.052±0.000
Traffic	0.168±0.037	0.165±0.004	0.140±0.002	0.133±0.001	0.172±0.001	**0.124±0.002**	0.134±0.001
Taxi	0.586±0.004	0.416±0.004	0.346±0.348	0.360±0.201	0.327±0.001	**0.314±0.003**	0.377±0.002
Wikipedia	0.379±0.004	0.247±0.001	1.549±1.017	0.236±0.000	0.333±0.001	0.282±0.002	**0.274±0.007**

Experiment/Results

Table 5: Test set MSE comparison (lower is better) of models from Salinas et al. (2019a) and our models LSTM-Real-NVP, LSTM-MAF and Transformer-MAF.

Data set	Vec-LSTM ind-scaling	Vec-LSTM lowrank-Copula	GP scaling	GP Copula	LSTM Real-NVP	LSTM MAF	Transformer MAF
Exchange	$\mathbf{1.6 \times 10^{-4}}$	1.9×10^{-4}	2.9×10^{-4}	1.7×10^{-4}	2.4×10^{-4}	3.8×10^{-4}	3.4×10^{-4}
Solar	9.3×10^{2}	2.9×10^{3}	1.1×10^{3}	9.8×10^{2}	$\mathbf{9.1 \times 10^{2}}$	9.8×10^{2}	9.3×10^{2}
Electricity	2.1×10^{5}	5.5×10^{6}	$\mathbf{1.8 \times 10^{5}}$	2.4×10^{5}	2.5×10^{5}	$\mathbf{1.8 \times 10^{5}}$	2.0×10^{5}
Traffic	6.3×10^{-4}	1.5×10^{-3}	5.2×10^{-4}	6.9×10^{-4}	6.9×10^{-4}	$\mathbf{4.9 \times 10^{-4}}$	5.0×10^{-4}
Taxi	7.3×10^{1}	5.1×10^{1}	2.7×10^{1}	3.1×10^{1}	2.6×10^{1}	$\mathbf{2.4 \times 10^{1}}$	4.5×10^{1}
Wikipedia	7.2×10^{7}	3.8×10^{7}	5.5×10^{7}	4.0×10^{7}	4.7×10^{7}	3.8×10^{7}	$\mathbf{3.1 \times 10^{7}}$

Experiment/Results

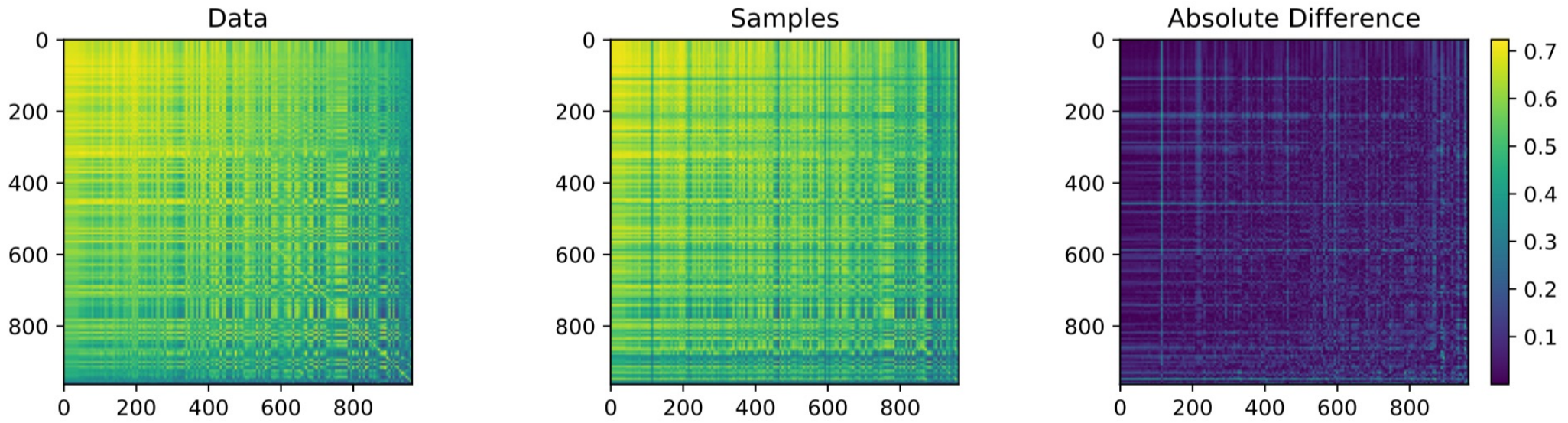


Figure 5: Visual analysis of the dependency structure extrapolation of the model. *Left*: Cross-covariance matrix computed from the test split of `Traffic` benchmark. *Middle*: Cross-covariance matrix computed from the mean of 100 sample trajectories drawn from the `Transformer-MAF` model's extrapolation into the future (test split). *Right*: The absolute difference of the two matrices mostly shows small deviations between ground-truth and extrapolation.

Results

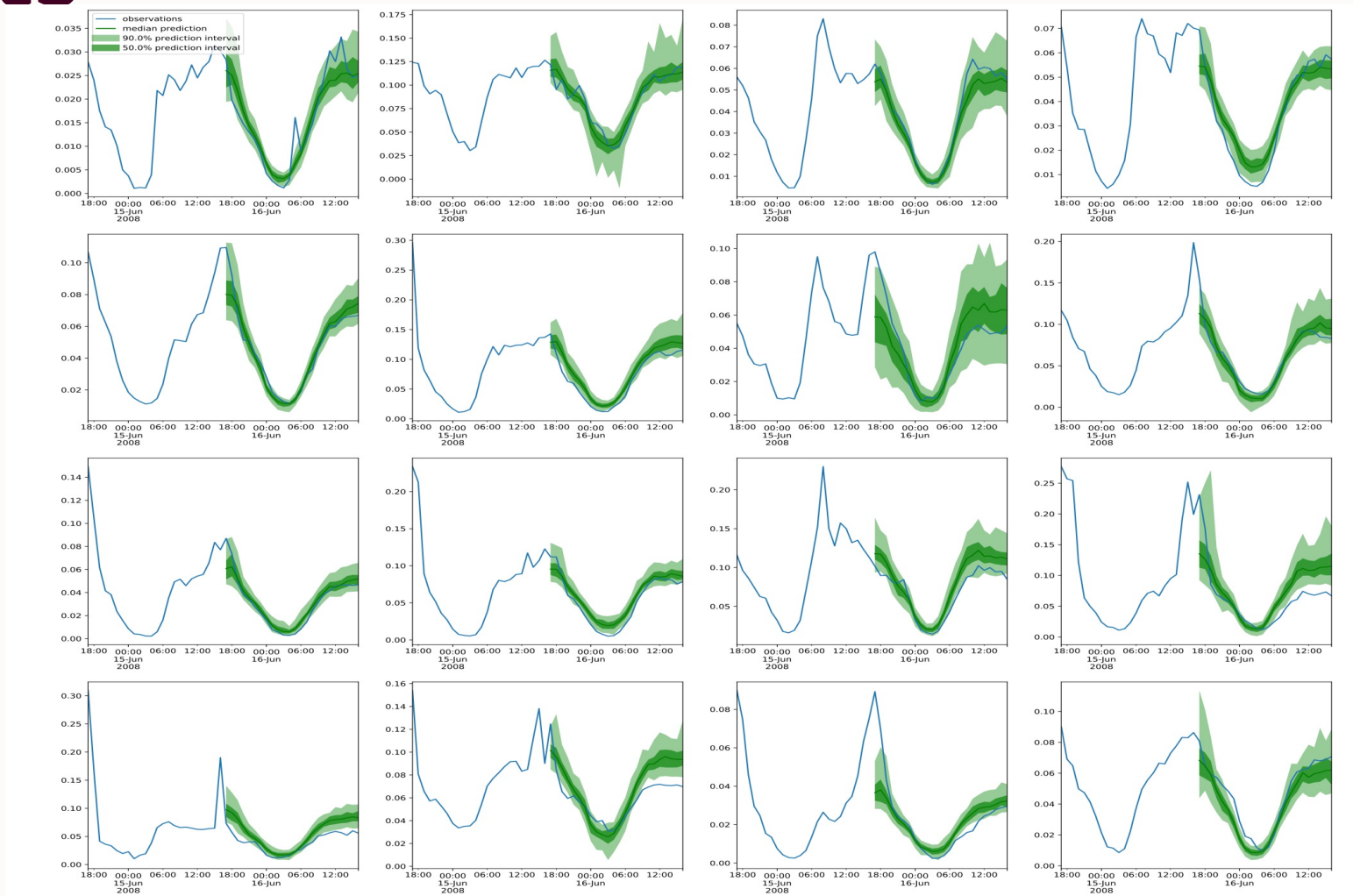


Figure 8: Prediction intervals and test set ground-truth from `LSTM-REAL-NVP` model for `Traffic` data of the first 16 of 963 time series.

Results

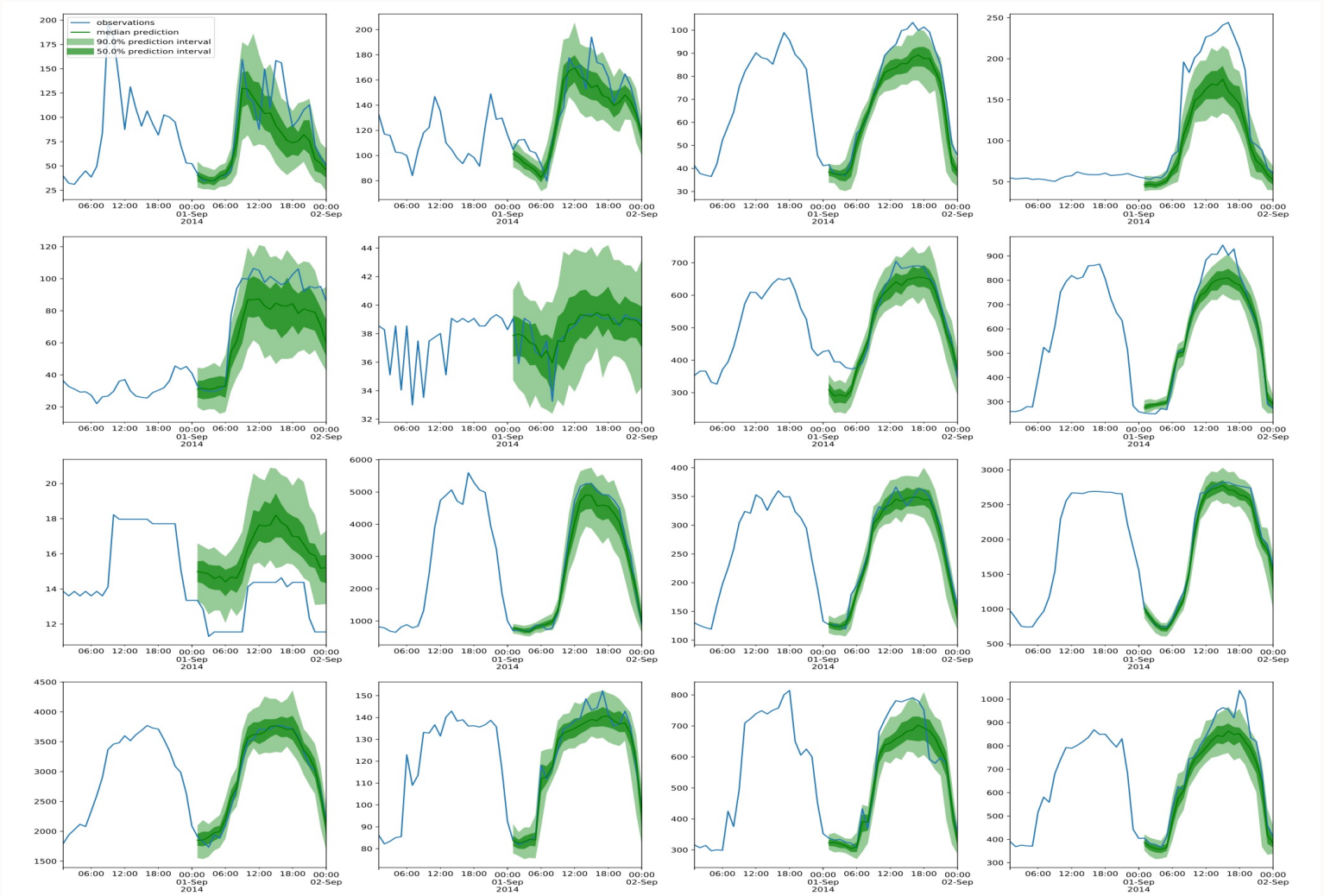


Figure 10: Prediction intervals and test set ground-truth from LSTM-REAL-NVP model for Electricity data of the first 16 of 370 time series.

Results

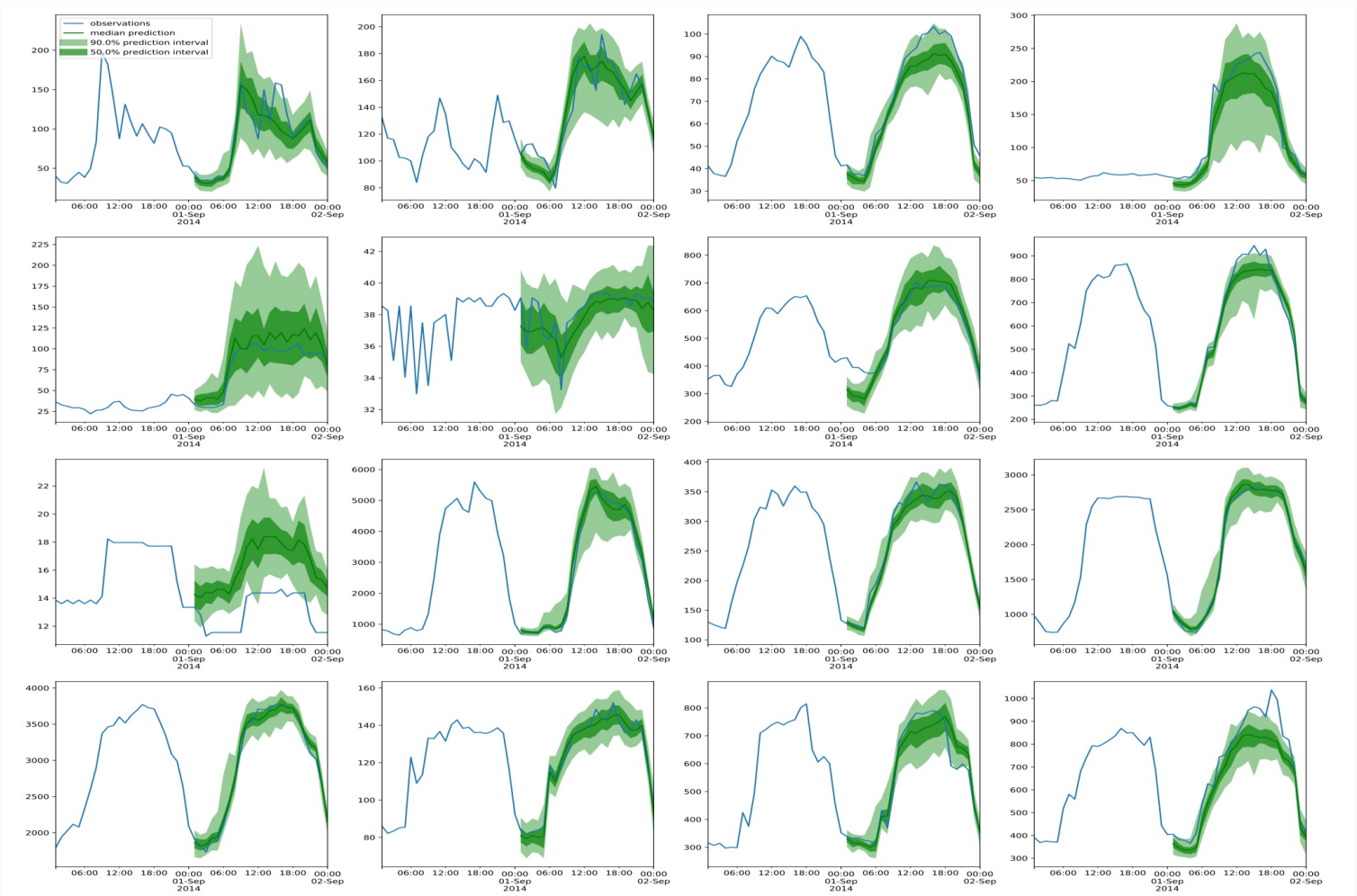


Figure 11: Prediction intervals and test set ground-truth from `Transformer-MAF` model for `Electricity` data of the first 16 of 370 time series.

Future Work

- Incorporate a better underlying flow model for e.g. trying using Flow++
- specific affine coupling layer
- More expressive conditioning

References

- Introduction to Normalizing flows: https://mbrubake.github.io/cvpr2021-nf_in_cv-tutorial/Introduction%20-%20CVPR2021.pdf
- A Multivariate Time Series Guide to Forecasting and Modeling : https://www.analyticsvidhya.com/blog/2018/09/multivariate-time-series-guide-forecasting-modeling-python-codes/
- DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks: https://arxiv.org/pdf/1704.04110.pdf
- Normalizing flows - theory and implementation: https://www.youtube.com/watch?v=6u7o5--aYNc&ab_channel=TinyVolt
- Attention Is All You Need: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- DENSITY ESTIMATION USING REAL NVP: https://arxiv.org/pdf/1605.08803.pdf
- Deep Learning Time Series Forcasting: https://github.com/Alro10/deep-learning-time-series